3

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

OIPE JC90
NOV 16 1999
PATENT & TRADEMARK OFFICE

In re Patent Application of

CHRISTOPOULOS et al.

Serial No. **09/394,428**

Filed: **September 13, 1999**

For: **DOWN SCALING OF IMAGES**

Atty. Ref.: **2466-35**

Group: **2713**

Examiner:

November 16, 1999

Assistant Commissioner for Patents
Washington, DC 20231

SUBMISSION OF PRIORITY DOCUMENTS

Sir:

It is respectfully requested that this application be given the benefit of the foreign filing date under the provisions of 35 U.S.C. §119 of the following, a certified copy of which is submitted herewith:

Application No.	Country of Origin	Filed
9700926-0	**Sweden**	**14 March 1997**
9703849-1	**Sweden**	**22 October 1997**

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: John R. Lastova

John R. Lastova
Reg. No. 33,149

JRL:mm
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

PRV

PATENT- OCH REGISTRERINGSVERKET

Patentavdelningen

OIPE JC90
NOV 16 1999
PATENT & TRADEMARK OFFICE

Intyg
Certificate

Härmed intygas att bifogade kopior överensstämmer med de handlingar som ursprungligen ingivits till Patent- och registreringsverket i nedannämnda ansökan.

This is to certify that the annexed is a true copy of the documents as originally filed with the Patent- and Registration Office in connection with the following patent application.

(71)	*Sökande* *Applicant (s)*	*Telefonaktiebolaget L M Ericsson, Stockholm SE*
(21)	*Patentansökningsnummer* *Patent application number*	*9700926-0*
(86)	*Ingivningsdatum* *Date of filing*	*1997-03-14*

Stockholm, 1999-09-28

För Patent- och registreringsverket
For the Patent- and Registration Office

Emma Johnsson

Emma Johnsson

Avgift
Fee 170:-

TECHNICAL FIELD

The present invention relates to a method and a device for computing the Discrete Cosine Transform (DCT) for applications such as image and video transcoding and scalable video coding.

BACKGROUND OF THE INVENTION AND PRIOR ART

It is reasonable to expect that in the future a wide range of quality video services like High Definition TV (HDTV) will be available together with Standard Definition TV (SDTV), and lower quality video services such as videophone and videoconference services. Multimedia documents containing video will most probably not only be retrieved over computer networks, but also over telephone lines, Integrated Services Digital Network (ISDN), Asynchronous Transfer Mode (ATM), or even mobile networks.

The transmission over several types of links or networks with different bit rates and varying traffic load will require an adaptation of the bit rate to the available channel capacity. A main constraint on the systems is that the decoding of any level below the one associated with the transmitted format should not need the complete decoding of the transmitted source.

In order to maximise the integration of these various quality video services, a single coding scheme which can provide an unlimited range of video services is desirable. Such a coding scheme would enable users of different qualities to communicate with each other. For example, a subscriber to only a lower quality video service should be capable of decoding and reconstructing a digitally transmitted higher quality video signal, albeit at the lower quality service level to which he subscribes. Similarly, a higher quality service subscriber should be capable of decoding and reconstructing a digitally transmitted lower quality video

signal although, of course, its subjective quality will be no better than its transmitted quality.

The problem therefore is associated with the way in which video will be transmitted to subscribers with different requirements (picture quality, processing power, memory requirements, resolution, bandwidth, frame rate, etc.). The following points summarise the requirements:

- satisfy users having different bandwidth requirements,
- satisfy users having different computational power,
- adapt frame rate, resolution and compression ratio according to user preferences and available bandwidth,
- adapt frame rate, resolution and compression ratio according to network abilities,
- short delay, and
- conform with standards, if required.

One solution to the problem of satisfying the different requirements of the receivers is the design of scalable bitstreams. In this form of scalability, there is usually no direct interaction between a transmitter and a receiver. Usually, the transmitter is able to make a bit stream which consists of various layers which can be used by receivers with different requirements in resolution, bandwidth, frame rate, memory or computational complexity. If new receivers are added which do not have the same requirements as the previous ones, then the transmitter has to be re-programmed to accommodate the requirements of the new receivers. Briefly, in bit stream scalability, the abilities of the decoders must be known in advance.

A different solution to the problem is the use of transcoders. A transcoder accepts a received data stream encoded according to a first coding scheme and outputs an encoded data stream encoded according

to a second coding scheme. If one had a decoder which operated according to a second coding scheme then such a transcoder would allow reception of the transmitted signal encoded according to the first coding scheme without modifying the original encoder.

One situation that usually appears especially in multiparty conferences is that a particular receiver has a different bandwidth ability and/or a different computational requirements. For example, in a multipoint communication with participants connected through ISDN and Public Switched Telephone Network (PSTN), the bandwidth can vary from 28.8 kbits/s (PSTN) to more than 128 kbits/s (ISDN). Since video transmitted at as high bit rates as 128 kbits/s can not be transferred in PSTN lines, video transcoding has to be implemented in the Multipoint Control Unit (MCU) or Gateway.

This transcoding might has to implement a spatial resolution of the video in order to fit into the bandwidth of a particular receiver. For example, an ISDN subscriber might be transmitting video in Common Intermediate Format (CIF) (288x352 pixels), while a PSTN subscriber might be able to receive video only in a Quad Common Intermediate Format (QCIF) (144x176). Another example is when a particular receiver does not have the computational power to decode at a particular resolution and therefore a reduced resolution video has to be transmitted to that receiver. Additionally, transcoding of HDTV to SDTV requires a resolution reduction.

For example, the transcoder could be used to convert a 128 kbit/s video signal in CIF format conforming to ITU-T standard H.261, from an ISDN video terminal for transmission to a 28.8 Kbit/s video signal in QCIF format over a telephone line using ITU-T standard H.263.

It should also be noted that many scalable video coding systems require both the use of 8x8 and 4x4 DCT. For example, in L.H. Kieu and K.N. Ngan, "Cell-loss concealment techniques for layered video codecs in an ATM network", *IEEE Trans. On Image Processing*, Vol. 3, No. 5, pp. 666 677, September 1994, a scalable video coding system is described in which the base layer has lower resolution compared to the enhancement layer. In that system, an 8x8 DCT is applied in each of the 8x8 blocks of the image and the enhancemant layer is compressed using the 8x8 DCT. The base layer uses the 4x4 out of the 8x8 DCTs of each block of the enhancement layer and it is compressed using only 4x4 DCTs. This however is not beneficial since a 4x4 DCT usually results in reduced performance compared to the 8x8 DCT and it also requires that encoders and decoders are able to handle 4x4 DCTs/IDCTs.

The traditional method of downsampling an image consists of two steps, see J. Bao, H. Sun, T.C. Poon, "HDTV down conversion decoder", *IEEE Trans. On Consumer Electronics*, Vol. 42, No. 3, pp. 402-410, August 1996. First the image is filtered by an anti-aliasing low pass filter. The filtered image is downsampled by a desired factor in each dimension. For a DCT-based compressed image, the above method implies that the compressed image has to be recovered to the spatial domain by inverse DCT and then undergo the procedure of filtering and downsampling. If the image is to be compressed and transmitted again, this require an extra forward DCT after the undersampling stage. This can be the case where the undersampling is taking place in a Multipoint Control Unit - MCU in order to satisfy the requirements and bandwidth of a particular receiver, or in scalable video coding scheme.

In a different method, that works in the compressed domain, both the operations of filtering and downsampling are combined in the DCT domain. This is done by cutting DCT coefficients of high frequencies and using the inverse DCT with a fewer number of DCT coefficients in order

to reconstruct the reduced resolution image. For example, one can use the 4x4 out of the 8x8 and perform the IDCT of these coefficients in order to reduce the resolution by a factor of 2 in each dimension. This does not result in significant compression gains and additionally it requires that receivers are able to handle 4x4 DCTs. Furthermore, this method results in significant amount of block edge effects and distortions, due to the poor approximations introduced by simply discarding higher order coefficients.

The above method would be more useful if one had 16x16 DCT blocks and kept the low frequency 8x8 DCT coefficients in order to obtain the downsampled image. However, most image and video compression standard methods like JPEG, H.261, MPEG1, MPEG2 and H.263 segment the images into rectangular blocks of size 8x8 pixels and apply the DCT in these blocks.

Therefore, only 8x8 DCTs are available. A way to compute the 16x16 DCT coefficients is to apply inverse DCT in each of the 8x8 blocks and reconstruct the image. Then the DCT in blocks of size 16x16 can be applied and the 8x8 out of the 16x16 DCTs coefficients of each block can be kept, if a resolution reduction by a factor of 2 in each dimension is required.

This, however, requires complete decoding (perform 8x8 IDCTs) and re-transforming by performing 16x16 DCTs (would require 16x16 DCT hardware). However, if one could compute the 8x8 out of the 16x16 DCT coefficients by using only 8x8 transformations, then this method would be faster and also perform better than the one that uses the 4x4 out of the 8x8. It would also mean that computation of DCTs of size 16x16 is avoided and reduced memory requirements are obtained.

Furthermore, US A 5 107 345 describes an adaptive DCT schemes used in coding. The schemes uses 2x2, 4x4, 8x8 and 16x16 DCTs in order to obtain a flexible bit rate which can be varied according to the available transmission capacity.

US A 5 452 104 describes an image compression method based on the scheme described in US 5 107 345.

SUMMARY

It is an object of the present invention to provide a method and a device which overcome the problems associated with the use of DCT of different sizes as outlined above.

This object and others are obtained by a method and a device for the computation of an N-point DCT by using only transforms of size N/2. The present invention also provides a direct computational algorithm for obtaining the DCT coefficients of a signal block taken from two adjacent blocks, i.e. it can be used for directly obtaining the N point DCT of an original sequence from 2 N/2 DCTs, which are representing the DCT coefficients for the first N/2 data points of the original sequence and the last N/2 data points of the original sequence, respectively.

Furthermore, a method that can be used for decreasing the spatial resolution of the incoming video is also obtained. The method provides lower spatial resolution reconstructed video with good picture quality, less complexity and memory requirements. It can be applied for image and/or video transcoding from a certain resolution factor to a lower one, while in the compressed domain. It can also be applied in scalable video coding and in adaptive video coding schemes. The main advantage of the scheme is that it requires DCT algorithms of standard size (8x8 in the

case of the existing video standards) and it results in better performance compared to existing schemes.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will now be described by way of non-limiting examples and with reference to the accompanying drawings, in which:

- Fig. 1 is a diagram illustrating a multipoint communication system.
- Fig. 2 is a flow chart, which shows the different steps carried out when transcoding a CIF image to QCIF in the DCT domain.
- Fig. 3 is a flow chart illustrating different steps carried out when transcoding a still image by reducing the resolution by a factor 2 in each dimension.
- Fig. 4 is a general view of a video transcoder.

DESCRIPTION OF PREFERRED EMBODIMENTS

In fig 1, a transmission system for digitised images is shown. Thus, in this example three users 101, 103 and 105 are connected to each other via an MCU 107. The users in this case have different capabilities. The users 101 and 105 are connected via 128 kbit/s ISDN connections, and the user 103 is connected via a 28.8 kbit/s PSTN connection. In a point-to-point communication, users 101 and 103 can also be connected through a gateway.

In such a case, the users 101 and 105 may transmit video signals in a CIF format to each other. However, if the user 103 wants to receive the video signal transmitted between the users 101 and 105, he/she is unable to do so, due to the limited transmission capacity of his/her transmission line, unless some kind of bit reduction is performed in the MCU.

One way of obtaining this bit reduction at the MCU is to extract the 4x4 low frequency coefficients of the 8x8 DCT coefficients of the incoming

video from the users 101 and 105 and to only transmit these to the user 103 in order to reconstruct the incoming frames in QCIF format through appropriate scaling of the motion vectors. This will not be beneficial from a compression and quality point of view. Instead, it would be more beneficial if low frequency 8x8 DCT coefficients were extracted from 16x16 blocks of DCT coefficients. This can then be performed in the following manner without having to use other DCTs/IDCTs than 8x8 DCTs.

Let the DCT coefficients of 4 adjacent 8x8 blocks of the CIF image be stored in 2D arrays in the form $Z = \begin{bmatrix} \Phi_1 & \Phi_2 \\ \Phi_3 & \Phi_4 \end{bmatrix}$ where Φ_i $(i = 1,2,3,4)$ are $(\frac{N}{2} x \frac{N}{2})$-point arrays (of DCT coefficients), where N = 16 in the following examples.

Each row k of Z consists of row k of block Φ_i and of row k of block Φ_j (i = 1 and j = 2 or i = 3 and j = 4). For each row k of Z, the problem now becomes to calculate the N point DCT when having the N/2 DCT points of Φ_i and Φ_j (i = 1 and j = 2 or i = 3 and j = 4).

In order to solve the problem of calculating the N point DCT from two N/2 DCT sequences, the following method can be used. Suppose that the sequence x_i, $i = 0,1,\dots,N\text{-}1$ is present. Then consider the following sequences: $y_i = x_i$, $i = 0,1,\dots,(N/2)\text{-}1$, and $z_i = x_{i+N/2}$, $i = 0,1,\dots,(N/2)-1$. Also assume that $N = 2^m$, and assume that hardware for the computation of the N/2-point DCT/IDCT is available in the MCU 107. In this specific case N = 16, which today is the normal case for computing DCT/IDCT since N/2 = 8, and 8x8 DCTs are mainly used in standard video coding schemes.

The problem is to compute the DCT coefficients of x_i by having the DCT coefficients of y_i and z_i. For an downsampling by a factor of 2, in this case half of the DCT coefficients of x_i (the low frequency coefficients) are needed.

First some necessary definitions are given.

Definitions

The normalised DCT (DCT-II) of x_i is given from the equation, see K.R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages and Applications*, Academic Press Inc., 1990:

$$X_k = \sqrt{\frac{2}{N}}\varepsilon_k \sum_{i=0}^{N-1} x_i \cos\frac{(2i+1)k\pi}{2N}, \quad k = 0,1,\ldots,N-1 \tag{1}$$

and the inverse DCT (IDCT) is given from the equation:

$$x_i = \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1} \varepsilon_k X_k \cos\frac{(2i+1)k\pi}{2N}, \; i = 0,1,\ldots,N-1 \tag{2}$$

where

$$\varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{for k = 0} \\ 1 & \text{for k} \neq 0 \end{cases} \tag{3}$$

Notice that $\varepsilon_{2k} = \varepsilon_k$ and $\varepsilon_{2k+1} = 1$.

The normalised DCT-IV of x_i is given from the equation, see the above cited book by K.R. Rao et al.

$$X_k = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} x_i \cos \frac{(2i+1)(2k+1)\pi}{4N}, \quad k = 0,1,\ldots,N-1 \tag{4}$$

and the inverse DCT-IV (IDCT-IV) is given from:

$$x_i = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X_k \cos \frac{(2k+1)(2i+1)\pi}{4N}, \quad i = 0,1,\ldots,N-1 \tag{5}$$

Notice that the DCT-IV and the IDCT-IV are given from the same equation.

The normalised DST-IV of x_i is given from the equation, see the above cited book by K.R. Rao et al.

$$X_k = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} x_i \sin \frac{(2i+1)(2k+1)\pi}{4N}, \quad k = 0,1,\ldots,N-1 \tag{6}$$

and the inverse DST-IV (IDST-IV) is given from:

$$x_i = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X_k \sin \frac{(2k+1)(2i+1)\pi}{4N}, \quad i = 0,1,\ldots,N-1 \tag{7}$$

Notice that the DST-IV and the IDST-IV are given from the same equation.

It should be noted that the normalisation factors $\sqrt{2/N}$ that appear in both the forward and inverse transforms can be merged as $2/N$ and move to either the forward or inverse transforms. In the following however the normalisation factor $\sqrt{2/N}$ will be kept in both the forward and the inverse transforms.

Furthermore, both the DST-IV and the DCT-IV can be computed through the DCT. In the above cited book by K.R. Rao et al, the software for the computation of the DCT-IV and the DST-IV through the DCT is given.

Suppose that the DCTs of y_i and z_i are denoted Y_k and Z_k respectively for $k = 0,1,\ldots,(N/2)-1$.

Two problems are addressed here:

(a) the computation of the N-point DCT of x_i by using only (N/2)-point transformations, and

(b) the computation of the N-point DCT of x_i when Y_k and Z_k are known (i.e. one has the DCT coefficients of the N/2-point sequences y_i and z_i).

Consider the even-indexed output of X_k.

From eq. (1), for $k = 2k$

$$
\begin{aligned}
X_{2k} &= \sqrt{\frac{2}{N}}\varepsilon_{2k}\sum_{i=0}^{N-1} x_i \cos\frac{(2i+1)2k\pi}{2N} \\
&= \sqrt{\frac{2}{N}}\varepsilon_k\left\{\sum_{i=0}^{\frac{N}{2}-1} x_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + \sum_{i=0}^{\frac{N}{2}-1} x_{i+N/2}\cos\frac{(2i+N+1)k\pi}{2\left(\frac{N}{2}\right)}\right\} \\
&= \sqrt{\frac{2}{N}}\varepsilon_k\left\{\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + \sum_{i=0}^{\frac{N}{2}-1} z_i\cos\left[\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + k\pi\right]\right\} \\
&= \sqrt{\frac{2}{N}}\varepsilon_k\left\{\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + (-1)^k\sum_{i=0}^{\frac{N}{2}-1} z_i\cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)}\right\} \\
&= \sqrt{\frac{1}{2}}\left\{\sqrt{\frac{2}{N/2}}\varepsilon_k\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + (-1)^k\sqrt{\frac{2}{N/2}}\varepsilon_k\sum_{i=0}^{\frac{N}{2}-1} z_i\cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)}\right\} \qquad (8) \\
&= \sqrt{\frac{1}{2}}\left[Y_k + (-1)^k Z_k\right] \qquad k = 0,1,\ldots,(N/2)-1.
\end{aligned}
$$

Equation (8) denotes that the even-indexed DCT coefficients of x_i can be computed by the DCT coefficients of y_i and z_i, i.e. the even indexed DCT coefficients of the N-element array can be obtained from the DCT coefficients of the two adjacent N/2 element arrays.

Then consider the odd-indexed DCT coefficients of X_k. For $k = 2k + 1$, eq. (1) becomes:

$$\begin{aligned}
X_{2k+1} &= \sqrt{\frac{2}{N}}\varepsilon_{2k+1}\sum_{i=0}^{N-1} x_i \cos\frac{(2i+1)(2k+1)\pi}{2N} \\
&= \sqrt{\frac{2}{N}}\left\{\sum_{i=0}^{\frac{N}{2}-1} x_i \cos\frac{(2i+1)(2k+1)\pi}{2N} + \sum_{i=0}^{\frac{N}{2}-1} x_{i+N/2} \cos\frac{(2i+N+1)(2k+1)\pi}{2N}\right\} \\
&= \sqrt{\frac{2}{N}}\left\{\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)(2k+1)\pi}{2N} + \sum_{i=0}^{\frac{N}{2}-1} z_i \cos\left[\frac{(2i+1)(2k+1)\pi}{2N} + (k\pi + \frac{\pi}{2})\right]\right\} \quad (9) \\
&= \sqrt{\frac{2}{N}}\left\{\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)(2k+1)\pi}{2N} + (-1)^{k+1}\sum_{i=0}^{\frac{N}{2}-1} z_i \sin\frac{(2i+1)(2k+1)\pi}{2N}\right\} \\
&= \sqrt{\frac{2}{N}}(X1_k - (-1)^k X2_k), \quad k = 0,1,\ldots,(N/2)-1.
\end{aligned}$$

Notice that $X1_k$ is the DCT-IV of y_i and $X2_k$ is the DST-IV of z_i. This means that X_{2k+1} can be computed through N/2 point transformations. Since the DCT-IV and the DST-IV can be computed through the DCT, this means that X_{2k+1} can be computed through a N/2 point DCT. From equation (8), X_{2k} can be computed through N/2 point DCTs and therefore an N-point DCT is not needed.

Below the terms $X1_k$ and $X2_k$ of equation (9) are analysed.

$$X1_k = \sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)(2k+1)\pi}{2N} =$$

$$= \sqrt{\frac{N/2}{2}}\sqrt{\frac{2}{N/2}}\sum_{i=0}^{\frac{N}{2}-1}\cos\frac{(2i+1)(2k+1)\pi}{2N}\left(\sqrt{\frac{2}{N/2}}\sum_{p=0}^{\frac{N}{2}-1}\varepsilon_p Y_p \cos\frac{(2i+1)p\pi}{2(N/2)}\right) \tag{10}$$

$k = 0,1,\ldots,(N/2)-1$

where by definition

$$y_i = \sqrt{\frac{2}{N/2}}\sum_{p=0}^{\frac{N}{2}-1}\varepsilon_p Y_p \cos\frac{(2i+1)p\pi}{2(N/2)} = IDCT^{II}_{N/2}(Y_p),\ \ i = 0,1,\ldots,(N/2)-1 \tag{11}$$

Therefore $X1_k$ can be computed by an IDCT followed by a forward DCT-IV of size N/2 (and multiplied by $\sqrt{\frac{2}{N/2}}$). Notice that the cos(.) terms in eq. (10), can be pre-computed and stored.

In a similar manner $X2_k$ can be calculated as:

$$X2_k = \sum_{i=0}^{\frac{N}{2}-1} z_i \sin\frac{(2i+1)(2k+1)\pi}{2N} =$$

$$= \sqrt{\frac{N/2}{2}}\left(\sqrt{\frac{2}{N/2}}\sum_{i=0}^{\frac{N}{2}-1}\sin\frac{(2i+1)(2k+1)\pi}{2N}\sqrt{\frac{2}{N/2}}\sum_{p=0}^{\frac{N}{2}-1}\varepsilon_p Z_p \cos\frac{(2i+1)p\pi}{2(N/2)}\right),\ \ k = 0,1,\ldots,(N/2)-1 \tag{12}$$

where by definition

$$z_i = \sqrt{\frac{2}{N/2}} \sum_{p=0}^{\frac{N}{2}-1} \varepsilon_p Z_p \cos\frac{(2i+1)p\pi}{2(N/2)} = IDCT^{II}_{N/2}(Z_p), \quad i = 0,1,\ldots,(N/2)-1 \qquad (13)$$

Therefore $X2_k$, can be computed by an inverse DCT followed by a forward DST-IV of size N/2 (and multiplied by $\sqrt{\frac{2}{N/2}}$). Notice that the cos(.) terms in eq. 12, can be pre-computed and stored.

Notice that in equations (10) and (12), a fast algorithm can be used for the computation of the DST-IV and DCT-IV as the one described in H.-C. Chiang and J.-C. Liu, "A progressive structure for on-line computation of arbitrary length DCT-IV and DST-IV transforms", *IEEE Trans. On Circuits and Systems for Video Technology*, Vol. 6, No. 6, pp. 692-695, Dec. 1996.

Alternatively, both the DCT-IV and the DST-IV can be computed through the DCT as explained in Z. Wang, "On computing the Discrete Fourier and Cosine Transforms", *IEEE Trans. On Acoustics, Speech and Signal Processing*, Vol. ASSP-33, No. 4, pp. 1341-1344, October 1985.

Therefore, a separate DCT-IV or DST-IV module is not required. DCT and IDCT is used only. Furthermore, for N = 16, a 16 point DCT is not required and the standard 8 point DCT can be used. This further reduces the complexity of the circuits required. Notice also that the cascaded operations of IDCT and DCT-IV (eq. 10) as well as IDCT and DST-IV (eq. 12) (all are of size N/2) can be replaced by a single N-point IDCT that can be used on a multiplexed basis, as described in N. R. Murthy and M. N. S. Swamy, "On a novel decomposition of the DCT and its applications", *IEEE Trans. On Signal Processing*, Vol. 41, No. 1, pp. 480-485, Jan. 1993.

This has certain advantages in hardware implementation of the algorithm. These equations therefore imply that standard available DCT hardware can be

used to compute the N-point DCT by having the DCT coefficients of the 2 adjacent blocks of N/2 points that constitute the N points.

The computational complexity of the algorithm depends on the algorithm used for the computation of the DCT and IDCT. The computational complexity appears to be similar to the complexity of a scheme that implements two inverse DCTs of size N/2 and a forward DCT of size N. However, such a scheme would require a N point DCT which is not advantageous, since it is supposed that N/2-point DCTs are available. Furthermore, the memory requirements are reduced in this scheme since an N-point DCT is not needed.

Notice that the above algorithms will compute all N DCT points. In practice this is not required for applications where image downsampling is performed. For example, for downsampling by a factor of 2 we have to keep the 8 out of every 16 DCT points of x_i. Therefore, $k = 0,1,\ldots,(N/4)-1$ in equations 8,9,10,12. Pruning DCT algorithms as in A.N. Skodras, "Fast Discrete Cosine Transform Pruning", *IEEE Trans. On Signal Processing*, Vol. 42, No. 7, pp. 1833-1837, July 1994, can be used in that case to compute only the required number of DCT points.

The equations given above can be further analysed and simplified. The detailed analysis follows below based on equation (9) and separate analysis of $X1_k$ and $X2_k$. Parts of equations derived in the previous paragraph are repeated for clarification purposes.

From equation (9)

$$
\begin{aligned}
X1_k &= \sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)(2k+1)\pi}{2N} = \\
&= \sum_{i=0}^{\frac{N}{2}-1} \cos\frac{(2i+1)(2k+1)\pi}{2N}\left(\sqrt{\frac{2}{N/2}}\sum_{p=0}^{\frac{N}{2}-1}\varepsilon_p Y_p \cos\frac{(2i+1)p\pi}{2(N/2)}\right) \\
&= \sum_{i=0}^{\frac{N}{2}-1} \cos\frac{(2i+1)(2k+1)\pi}{2N} \\
&\qquad \sqrt{\frac{2}{N/2}}\left\{\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_{2p} Y_{2p} \cos\frac{(2i+1)p\pi}{2(N/4)} + \sum_{p=0}^{\frac{N}{4}-1}\varepsilon_{2p+1} Y_{2p+1} \cos\frac{(2i+1)(2p+1)\pi}{2(N/2)}\right\}
\end{aligned} \tag{14}
$$

By defining the sequences $Y1$ and $Y2$ as:

$$
\begin{aligned}
Y1_p &= Y_{2p} \\
Y2_p &= Y_{2p+1}
\end{aligned} \quad \text{for } p = 0,1,\ldots,N/4 \tag{15}
$$

equation (14) becomes

$$
\begin{aligned}
X1_k &= \sum_{i=0}^{\frac{N}{2}-1} \cos\frac{(2i+1)(2k+1)\pi}{2N} \\
&\qquad \sqrt{\frac{2}{N/2}}\left\{\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Y1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sum_{p=0}^{\frac{N}{4}-1} Y2_p \cos\frac{(2i+1)(2p+1)\pi}{2(N/2)}\right\} \\
&= \sum_{i=0}^{\frac{N}{2}-1} \cos\frac{(2i+1)(2k+1)\pi}{2N} \\
&\qquad \frac{1}{\sqrt{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Y1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Y2_p \cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\}
\end{aligned} \tag{16}
$$

Equation (16) can be subdivided further into:

$$X1_k = \sum_{i=0}^{\frac{N}{4}-1} \cos\frac{(2i+1)(2k+1)\pi}{2N}$$

$$\frac{1}{\sqrt{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Y1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Y2_p \cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\}$$

$$+ \tag{17}$$

$$\sum_{i=N/4}^{\frac{N}{2}-1} \cos\frac{(2i+1)(2k+1)\pi}{2N}$$

$$\frac{1}{\sqrt{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Y1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Y2_p \cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\}$$

By defining

$$y1_i = \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Y1_p \cos\frac{(2i+1)p\pi}{2(N/4)}, \quad i = 0,1,\ldots,(N/4)-1 \tag{18}$$

and

$$y2_i = \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Y2_p \cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}, \quad i = 0,1,\ldots,(N/4)-1 \tag{19}$$

it is seen that $y1_i$ is the IDCT of $Y1_p$ of N/4 points and $y2_i$ is the IDCT-IV of $Y2_p$ of N/4 points.

Notice that when $Y1$ and/or $Y2$ are zero, then $y1_i$ and/or $y2_i$ do not need to be computed. This will speed-up the calculation of equation (17).

Further analysis of the second term of equation (17) gives:

$$\sum_{i=N/4}^{\frac{N}{2}-1} \cos\frac{(2i+1)(2k+1)\pi}{2N}$$
$$\frac{1}{\sqrt{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Y1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Y2_p \cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\}$$

$$=$$

$$\sum_{i=0}^{\frac{N}{4}-1} \cos\frac{(2i+1+\frac{N}{2})(2k+1)\pi}{2N}$$
$$\frac{1}{\sqrt{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Y1_p \cos\frac{(2i+1+\frac{N}{2})p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Y2_p \cos\frac{(2i+1+\frac{N}{2})(2p+1)\pi}{4(N/4)}\right\}$$

$$=$$

$$\sum_{i=0}^{\frac{N}{4}-1} \cos\frac{(2i+1+\frac{N}{2})(2k+1)\pi}{2N}$$
$$\frac{1}{\sqrt{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p (-1)^p Y1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} (-1)^{p+1} Y2_p \sin\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\}$$

(20)

By defining

$$y1_i^{*} = \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p (-1)^p Y1_p \cos\frac{(2i+1)p\pi}{2(N/4)}, \quad i = 0,1,\ldots,(N/4)-1 \qquad (21)$$

$$y2_i^{*} = \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}(-1)^{p+1} Y2_p \sin\frac{(2i+1)(2p+1)\pi}{4(N/4)}, \quad i = 0,1,\ldots,(N/4)-1 \qquad (22)$$

$y1_i^{*}$ is recognised as the IDCT of sequence $(-1)^p Y1_p$ of N/4 points and $y2_i^{*}$ is recognised as the IDST-IV of sequence $(-1)^{p+1} Y2_p$, of N/4 points.

Notice that when $Y1$ and/or $Y2$ are zero, then $y1_i'$ and/or $y2_i'$ do not need to be computed. This will speed-up the calculation of equation (20).

From equations 10, 11, 13 and 14, it is seen that

$$X1_k = \frac{1}{\sqrt{2}}\left\{\sum_{i=0}^{\frac{N}{4}-1}\cos\frac{(2i+1)(2k+1)\pi}{2N}\quad(y1_i' + y2_i') \quad + \sum_{i=0}^{\frac{N}{4}-1}\cos\frac{(2i+1+\frac{N}{2})(2k+1)\pi}{2N}\quad(y1_i'' + y2_i'')\right\}$$

$k=0,1,\ldots,(N/2)-1$

(23)

In similar manner, the second term of equation (9), can be analysed as follows:

$$X2_k = \sum_{i=0}^{\frac{N}{2}-1} z_i \sin\frac{(2i+1)(2k+1)\pi}{2N} =$$

$$= \sum_{i=0}^{\frac{N}{2}-1}\sin\frac{(2i+1)(2k+1)\pi}{2N}\sqrt{\frac{2}{N/2}}\sum_{p=0}^{\frac{N}{2}-1}\varepsilon_p Z_p \cos\frac{(2i+1)p\pi}{2(N/2)}$$

$$= \sum_{i=0}^{\frac{N}{2}-1}\sin\frac{(2i+1)(2k+1)\pi}{2N}$$

$$\sqrt{\frac{2}{N/2}}\left\{\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_{2p}Z_{2p}\cos\frac{(2i+1)p\pi}{2(N/4)} + \sum_{p=0}^{\frac{N}{4}-1}\varepsilon_{2p+1}Z_{2p+1}\cos\frac{(2i+1)(2p+1)\pi}{2(N/2)}\right\}$$

$$= \sum_{i=0}^{\frac{N}{2}-1}\sin\frac{(2i+1)(2k+1)\pi}{2N}$$

$$\sqrt{\frac{1}{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Z1_p\cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}Z2_p\cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\} \quad (24)$$

where

$$Z1_p = Z_{2p} \quad \text{for } p = 0,1,\ldots,N/4$$
$$Z2_p = Z_{2p+1}$$

(25)

Equation (24) can be further subdivided to:

$$X2_k = \sum_{i=0}^{\frac{N}{4}-1} \sin\frac{(2i+1)(2k+1)\pi}{2N} \sqrt{\frac{1}{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Z1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Z2_p \cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\}$$
$$+ \sum_{i=N/4}^{\frac{N}{2}-1} \sin\frac{(2i+1)(2k+1)\pi}{2N} \sqrt{\frac{1}{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Z1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Z2_p \cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\} \qquad (26)$$

By defining

$$z1_i' = \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Z1_p \cos\frac{(2i+1)p\pi}{2(N/4)}, \quad i = 0,1,\ldots,(N/4)-1 \qquad (27)$$

$$z2_i' = \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Z2_p \cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}, \quad i = 0,1,\ldots,(N/4)-1 \qquad (28)$$

It is seen that $z1_i'$ is the IDCT of sequence $Z1_p$ of N/4 points and $z2_i'$ is the IDCT -IV of sequence $Z2_p$ of N/4 points

Notice that when $Z1_p$ and/o $Z2_p$ are zero, then $z1_i'$ and/or $z2_i'$ do not need to be computed. This will speed-up the calculation of equation (29).

Further analysis of the second term of equation (26) gives:

$$\sum_{i=N/4}^{\frac{N}{2}-1} \sin\frac{(2i+1)(2k+1)\pi}{2N}$$
$$\frac{1}{\sqrt{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Z1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Z2_p \cos\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\}$$

$$=$$

$$\sum_{i=0}^{\frac{N}{4}-1} \sin\frac{(2i+1+\frac{N}{2})(2k+1)\pi}{2N}$$
$$\frac{1}{\sqrt{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p Z1_p \cos\frac{(2i+1+\frac{N}{2})p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} Z2_p \cos\frac{(2i+1+\frac{N}{2})(2p+1)\pi}{4(N/4)}\right\}$$

$$=$$

$$\sum_{i=0}^{\frac{N}{4}-1} \sin\frac{(2i+1+\frac{N}{2})(2k+1)\pi}{2N}$$
$$\frac{1}{\sqrt{2}}\left\{\sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p (-1)^p Z1_p \cos\frac{(2i+1)p\pi}{2(N/4)} + \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1} (-1)^{p+1} Z2_p \sin\frac{(2i+1)(2p+1)\pi}{4(N/4)}\right\}$$

(29)

By defining

$$z1_i^{*} = \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}\varepsilon_p (-1)^p Z1_p \cos\frac{(2i+1)p\pi}{2(N/4)}, \quad i = 0,1,\ldots,(N/4)-1 \tag{30}$$

$$z2_i^{*} = \sqrt{\frac{2}{N/4}}\sum_{p=0}^{\frac{N}{4}-1}(-1)^{p+1} Z2_p \sin\frac{(2i+1)(2p+1)\pi}{4(N/4)}, \quad i = 0,1,\ldots,(N/4)-1 \tag{31}$$

It is seen that $z1''_i$ is the IDCT of sequence $(-1)^p Z1_p$ of N/4 points and $z2''_i$ is the IDST -IV of sequence $(-1)^{p+1} Z2_p$ of N/4 points. Notice that when $Z1_p$ and/or $Z2_p$ are zero, then $z1''_i$ and/or $z2''_i$ do not need to be computed. This will speed-up the calculation of equation (29).

From equations 26, 27, 28, 29, 30 and 31 it is seen that:

$$X2_k = \frac{1}{\sqrt{2}}\left\{\sum_{i=0}^{\frac{N}{4}-1}\sin\frac{(2i+1)(2k+1)\pi}{2N}\,(z1'_i + z2'_i) + \sum_{i=0}^{\frac{N}{4}-1}\sin\frac{(2i+1+\frac{N}{2})(2k+1)\pi}{2N}\,(z1''_i + z2''_i)\right\}$$

(32)

Therefore, the odd indexed DCT coefficients can be computed from equation

$$X_{2k+1} = \sqrt{\frac{2}{N}}(X1_k - (-1)^k X2_k), \quad k = 0,1,\ldots,(N/2)-1. \qquad (33)$$

Notice that in eq. (8) and (33), the values of k will be $k = 0,1,\ldots,(N/4)-1$. for a downsampling by a factor of 2.

Thus, for example, an image of QCIF format can be derived from an image in a CIF format without having to use any other transforms than 8x8 DCTs (if the CIF image had been processed by using DCT applied in 8x8 blocks) by using the following method illustrated in the flow chart in fig. 2.

First in block 201 four 8x8 adjacent DCT-point arrays of a CIF format image are loaded into a memory as an array of size 16x16 points. Next, the 16-point DCT for each row of the 16x16 array is calculated in a block

203 using the equations (8) and (9) for the even and odd coefficients, respectively. Then, the coefficients of that row are stored in a memory 205.

Thereupon it is checked in a block 207 if the current row was the last in the 16x16 array. If this is not the case the row number is incremented in a block 209 and the calculations in block 203 are repeated for the rext row of the 16x16 array. If, on the other hand, the 16 DCT coefficients for the last row have been calculated and stored in the memory, a block 211 fetches the 16x16 DCT coefficients now stored in the memory 205 and loads these into the block 211.

The procedure then continues in a similar manner for the computation of the columns, i.e. the method is applied in a column manner to the result that has been obtained from the row-computation.

Hence, in a block 213 the DCT for the first column of the array loaded into the block 211 is calculated using the equations (8) and (9) for the even and odd coefficients, respectively, and the coefficients for that column are stored in a block 215. Thereupon, it is checked in a block 217 if the DCT for the column currently calculated is the last that is required. If this is not the case the column number is incremented by one in a block 219 for the next column of the 16x16 array and the calculations in block 213 are repeated for the next column of the 16x16 array.

If, on the other hand, the 16 DCT coefficients for the last column have been calculated and stored in the memory block 215, a block 221 fetches the 16x16 DCT coefficients stored in the memory 215 and loads these into the block 221.

Next, in the block 221, the 8x8 low frequency DCT coefficients are extracted from the 16x16 DCT coefficients. The 8x8 DCT coefficients are then output in a block 223.

If only the MxK (M rows and K columns) DCT coefficients are required then the computation of the rows remains the same but then for each row, only the first K coefficients are computed. Then, during the computation of the columns, the first K columns are processed and for each of these columns the low frequency M coefficients are calculated. This method is useful for undersampling by a different factor in each dimension (for example undersampling by 2 in dimension x and by 4 in dimension y). Thereafter the MxK low frequency coefficients of the in this manner obtained 16x16-point DCT are extracted and transmitted. The method can also be applied in a similar manner to compute arbitrary number of DCT coefficients for each row/column.

The method can be used in a number of different applications. As an example, suppose that an image compression scheme like JPEG, uses 8x8 DCTs. Suppose that the compressed image is received. An undersampling (downsampling) of the image by a factor of 2 in each dimension would require keeping the low frequency 8x8 DCT coefficients out of a block of 16x16 DCT coefficients. These 16x16 DCT blocks can be computed with the method described above by having the 4(8x8) DCT coefficients that constitute the 16x16 block.

Notice that in the Row-Column (RC) computation, a further speed-up can be obtained if the coefficients of a certain row/column are zero, which normally is the case for high frequency DCT coefficients. In practice, in video coding about 80% of DCT coefficients are zero, i.e. the ones corresponding to high frequencies. Therefore, faster computation can be achieved by taking this information into account. For example, if all DCT coefficients of the two sub-rows of the fourth row of Z are zero, there is

no reason to try to compute the DCT coefficients for that row. Another case can for example be if the DCT coefficients of row 3 of Φ_2 are zero, all computations involving these coefficients can then be skipped.

Notice that the scheme can be applied in a recursive manner. For example, if QCIF, CIF and SCIF are required then 8x8 DCTs are used for the SCIF. The CIF is obtained by calculating the 8x8 DCTs of the 16x16 block that consists of 4(8x8) DCT coefficients of the SCIF. Then the QCIF can be obtained by keeping only the 4x4 out of the 8x8 DCT coefficients of each 8x8 block of the CIF or by again calculating the 8x8 DCTs of the 16x16 block that consists of 4(8x8) DCT coefficients of the CIF. This has interesting applications in scalable image/video coding schemes and in image/video transcoding with spatial resolution reduction schemes.

Alternatively, from each 8x8 blocks of DCT coefficients, one can keep only the 4x4 low frequency coefficients. Then from 4(4x4) blocks of DCT coefficients one can compute an 8x8 block of DCT coefficients.

The method as described herein has a number of advantages. Thus, standard DCT/IDCT hardware can be used, since there is no requirement of using 16x16 DCT, when 8x8 DCT/IDCT is available.

There is no requirement for fully decoding, filtering and downsampling in the spatial domain and fully encoding by DCT again. There are less memory requirements, since computation of a 16x16 DCT requires much more memory and data transfers compared to the 8x8 case.

The method can be used for undersampling by various factors. For example, if 8x8 DCTs are used and an undersampling by a factor of 4 in each dimension is desired, then only the low frequency 2x2 DCT coefficients out of the 8x8 are to be kept, which is not advantageous

from a compression efficiency point of view. However, with the method as described herein one can calculate the 16x16 DCT coefficients out of the available 4(8x8) DCTs and keep only the 4x4 of them, or compute them directly. This is more efficient than by keeping the 2x2 out of the 4x4 and will result in better image quality. One can also compute an 8x8 block of DCT coefficients by 4(4x4) blocks of DCT coefficients. Each of the 4x4 blocks of DCT coefficients can be part of an 8x8 block of DCT coefficients.

The method results in fast computation when many of the DCT coefficients of the 8x8 blocks are zero, since computation of rows and columns DCTs/IDCT's (type II or IV) and DST/IDST (type IV) can be avoided for that row/column.

Further, in L.H. Kieu and K.N. Ngan, "Cell-loss concealment techniques for layered video codecs in an ATM network", *IEEE Trans. On Image Processing*, Vol. 3, No. 5, pp. 666-677, September 1994, a frequency scalable video coding scheme is described. The scheme uses 8x8 DCTs for the upper layers. The base layer is coded using 4x4 DCTs. The low frequency 4x4 DCT coefficients of each of the 8x8 blocks of the upper layer are used at the base layer.

With the DCT algorithms as described herein, the frequency scalable video codec described in the above cited paper by L.H. Kieu et al. can be modified as follows:

- Compute the low-frequency 8x8 DCT coefficients by applying the proposed algorithm in 4(8x8) blocks of DCT coefficients of the upper layer. Then code the base layer by standard techniques using 8x8 DCT algorithms. This as an efficient technique for all frequency scalable systems. The method has the following advantages in this case:

The video coding is applied in 8x8 blocks. This results in better coding efficiency compared to using 4x4 blocks. The motion vectors have to be computed for 8x8 blocks. Therefore less motion vectors need to be transmitted (or stored) compared to using 4x4 blocks. Also, variable length coding schemes are well studied for 8x8 DCT coefficients compared to the 4x4 case.

Notice that an alternative method would be to keep the 4x4 low frequency DCT coefficients of each 8x8 DCT block of the upper layer and by having 4(4x4) of these blocks to compute the 8x8 DCT of these 4x4 blocks. Such an approach is illustrated in fig. 3.

Thus, in fig. 3 a flow chart illustrating different steps carried out when transcoding a still image by reducing the resolution by a factor 2 in each dimension, is shown. First in a block 301 an image compressed in the DCT domain is received. The received image is then entropy decoded in a block 303, for example by a Huffman decoder or an arithmetic decoder.

Thereupon, in a block 305, 8x8 blocks of DCT coefficients of the decoded full size image are obtained, and in a block 307 the low-frequency 4x4 DCT coefficients from each 8x8 block are extracted. 8x8 DCTs are then obtained in a block 309 by means of applying the row-column method described above for four adjacent 4x4 blocks of low-frequency coefficients.

Next, each 8x8 blocks resulting from the row-column method in the block 309 is entropy coded in a block 311 and then transmitted or stored in a block 313. Notice that the DCT coefficients might have to re-quantized before entropy coding in order to achieve a specific compression factor.

In fig. 4 a general view of a video transcoder employing the teachings of the method described above, is shown. The transcoder receives an incoming bitstream of a compressed video signal. The received compressed video signal is decoded in a block 401 wherein the motion vectors of the decompressed video signal are extracted. The motion vectors are fed to a block 403 in which a proper motion vector scaling in accordance with the transcoding performed by the transcoder is executed, as for example in this case a division by 2 is performed. The image information not relating to the motion vectors are fed to a block 405 from the block 401.

In the block 405 DCT blocks of size 8x8 are obtained. The DCT blocks of size 8x8 are then fed to a block 407 in which four adjacent 8x8 DCT blocks are combined to one, undersampled, 8x8 DCT block according to the method described above. The new, undersampled, 8x8 DCT blocks are then available in a block 409. A block 411 then encodes the 8x8 DCT blocks in the block 409 (this might also involve re-quantization of the DCT coefficients) together with the scaled motion vectors from the block 403 and forms a combined compressed output video signal.

Furthermore, in US A 5,107,345 and US A 5,452,104 an adaptive block size image compression method and system is proposed. For a block size of 16x16 pixels, the system calculates DCTs for the 16x16 blocks and the 8x8, 4x4 and 2x2 blocks that make the 16x16 block. The algorithm as described herein can be used to compute the NxN block by having the 4(N/2 x N/2) DCT coefficients. For example, by having the DCT coefficients of each 2x2 block one can compute the DCT coefficients for the 4x4 blocks. By having the DCT coefficients for each 4x4 block one can compute the DCT coefficients for the 8x8 blocks, etc. The DCT algorithm can therefore be used for the efficient coding in the schemes described in US A 5,107,345 and US A 5,452,104.

CLAIMS

1. A device for calculating the DCT for an original sequence of length N, N being a positive, even integer,
characterised by
- means for calculating the DCT directly from two sequences of length N/2 representing the first and second half of the original sequence, respectively, only using DCTs of length N/2.

2. A device for calculating the DCT for a sequence of length N, N being a positive, even integer, characterised by
- means for calculating the DCT directly from two DCTs of length N/2 representing the DCTs for the first and second half of the sequence, respectively.

3. A device for calculating the DCT for a sequence of length NxN, N being a positive, even integer, characterised by
- means for calculating the NxN DCT directly from four DCTs of length (N/2xN/2) representing the DCTs of four adjacent blocks constituting the NxN block.

4. A method of transcoding in the compressed (DCT) domain, wherein the compressed frames are undersampled by a certain factor in each dimension, characterised in that an NxN DCT is directly calculated from 4 adjacent N/2xN/2 blocks of DCT coefficients of the incoming compressed frames, N being a positive, even integer.

5. A method of calculating the DCT for an original sequence of length N, N being a positive, even integer,
characterised in that the DCT is calculated directly from two sequences of length N/2 representing the first and second half of the original sequence, respectively, only using DCTs of length N/2.

6. A method of calculating the DCT for a sequence of length N, N being a positive, even integer,
characterised in that the DCT is calculated directly from two DCTs of length N/2 representing the DCTs for the first and second half of the sequence, respectively.

7. A device for calculating DCTs of length N, where N is a positive even integer, characterised by
- means in the device for calculating DCTs of length N/2, arranged to calculate the even coefficients of a DCT of length N as:

$$X_{2k} = \sqrt{\frac{2}{N}}\varepsilon_{2k}\sum_{i=0}^{N-1} x_i \cos\frac{(2i+1)2k\pi}{2N}$$

$$= \sqrt{\frac{2}{N}}\varepsilon_k\left\{\sum_{i=0}^{\frac{N}{2}-1} x_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + \sum_{i=0}^{\frac{N}{2}-1} x_{i+N/2}\cos\frac{(2i+N+1)k\pi}{2\left(\frac{N}{2}\right)}\right\}$$

$$= \sqrt{\frac{2}{N}}\varepsilon_k\left\{\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + \sum_{i=0}^{\frac{N}{2}-1} z_i \cos\left[\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + k\pi\right]\right\}$$

$$= \sqrt{\frac{2}{N}}\varepsilon_k\left\{\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + (-1)^k\sum_{i=0}^{\frac{N}{2}-1} z_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)}\right\}$$

$$= \sqrt{\frac{1}{2}}\left\{\sqrt{\frac{2}{N/2}}\varepsilon_k\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)} + (-1)^k\sqrt{\frac{2}{N/2}}\varepsilon_k\sum_{i=0}^{\frac{N}{2}-1} z_i \cos\frac{(2i+1)k\pi}{2\left(\frac{N}{2}\right)}\right\},$$

$$= \sqrt{\frac{1}{2}}\left[Y_k + (-1)^k Z_k\right] \qquad k = 0,1,\ldots,(N/2)-1.$$

and the odd coefficients as:

$$
\begin{aligned}
X_{2k+1} &= \sqrt{\frac{2}{N}}\varepsilon_{2k+1}\sum_{i=0}^{N-1} x_i \cos\frac{(2i+1)(2k+1)\pi}{2N} \\
&= \sqrt{\frac{2}{N}}\left\{\sum_{i=0}^{\frac{N}{2}-1} x_i \cos\frac{(2i+1)(2k+1)\pi}{2N} + \sum_{i=0}^{\frac{N}{2}-1} x_{i+N/2}\cos\frac{(2i+N+1)(2k+1)\pi}{2N}\right\} \\
&= \sqrt{\frac{2}{N}}\left\{\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)(2k+1)\pi}{2N} + \sum_{i=0}^{\frac{N}{2}-1} z_i \cos\left[\frac{(2i+1)(2k+1)\pi}{2N} + (k\pi + \frac{\pi}{2})\right]\right\}. \\
&= \sqrt{\frac{2}{N}}\left\{\sum_{i=0}^{\frac{N}{2}-1} y_i \cos\frac{(2i+1)(2k+1)\pi}{2N} + (-1)^{k+1}\sum_{i=0}^{\frac{N}{2}-1} z_i \sin\frac{(2i+1)(2k+1)\pi}{2N}\right\} \\
&= \sqrt{\frac{2}{N}}(X1_k - (-1)^k X2_k), \quad k = 0,1,\ldots,(N/2)-1
\end{aligned}
$$

ABSTRACT

In a method and a device for calculation of the Discrete Cosine Transform (DCT) only the DCT coefficients representing the first half and the second half of an original sequence are required for obtaining the DCT for the entire original sequence. The device and the method is therefore very useful when calculation of DCTs of a certain length is supported by hardware and/or software, but when DCTs of other sizes are desired. Areas of application are for example still image and video transcoding, as well as scalable image and/or video coding.

(Fig. 2)

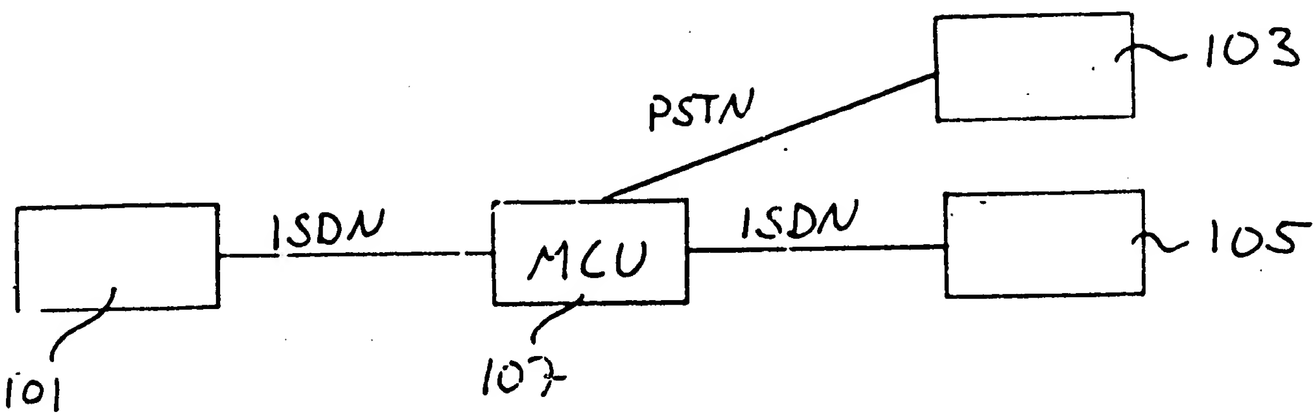

Fig. 1

PRV 97-03-14

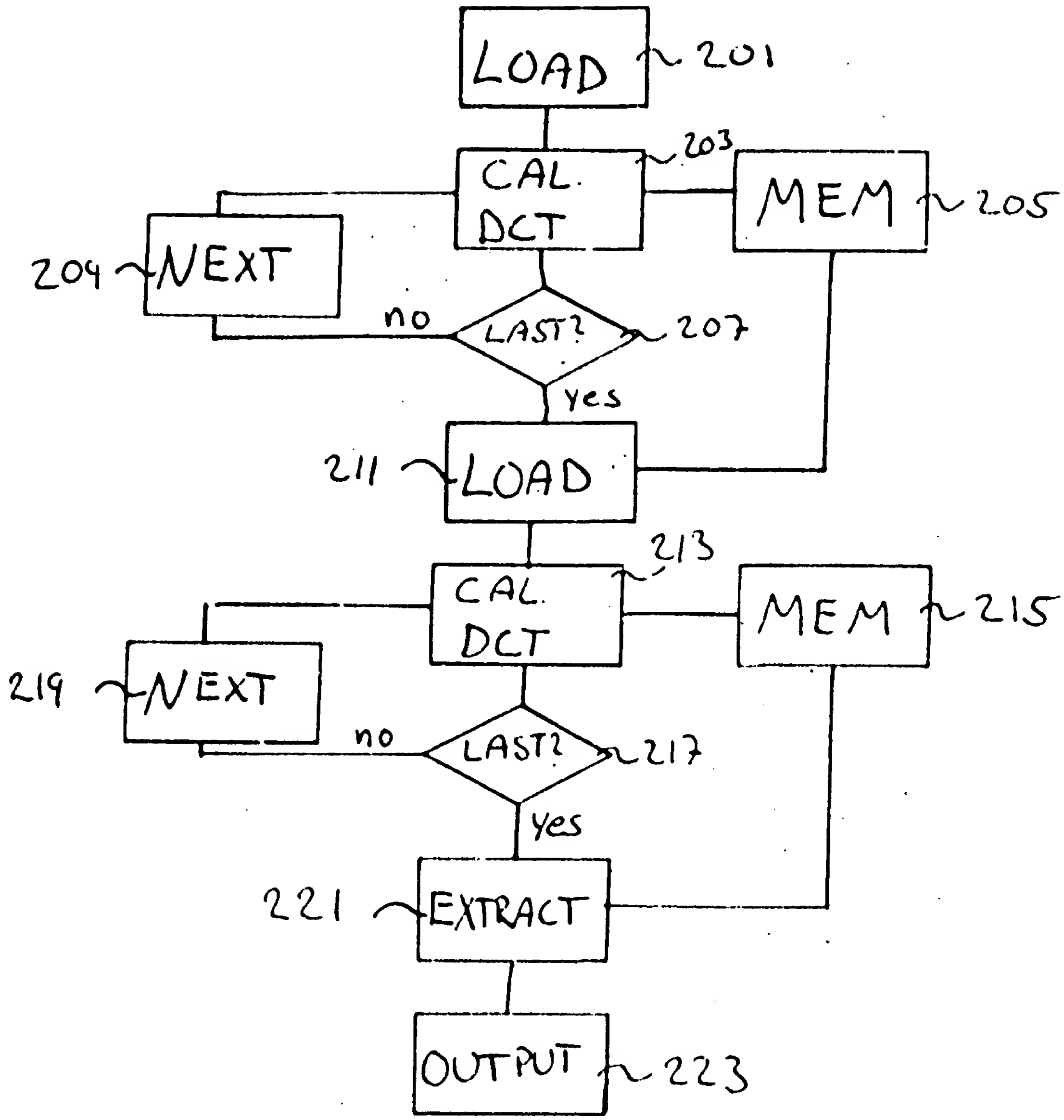

Fig. 2.

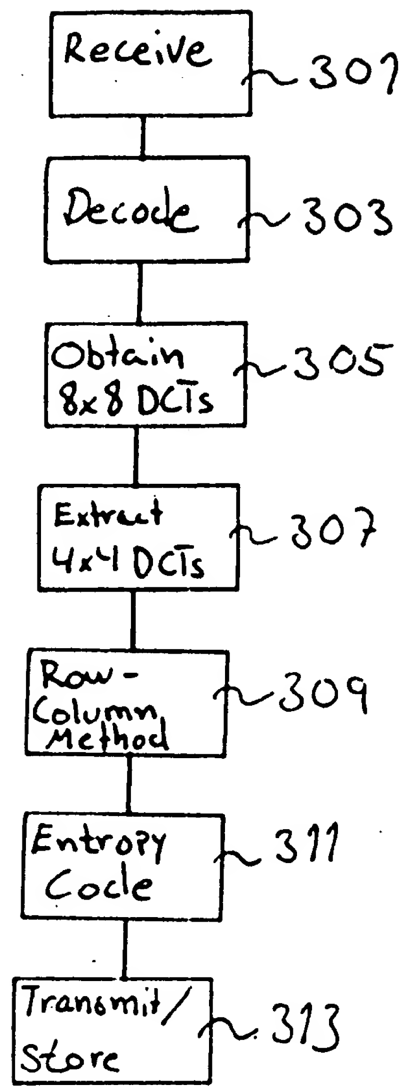
Receive
301
Decode
303
Obtain 8x8 DCTs
305
Extract 4x4 DCTs
307
Row-Column Method
309
Entropy Code
311
Transmit/ Store
313

Fig. 3

PRV 97-03-14

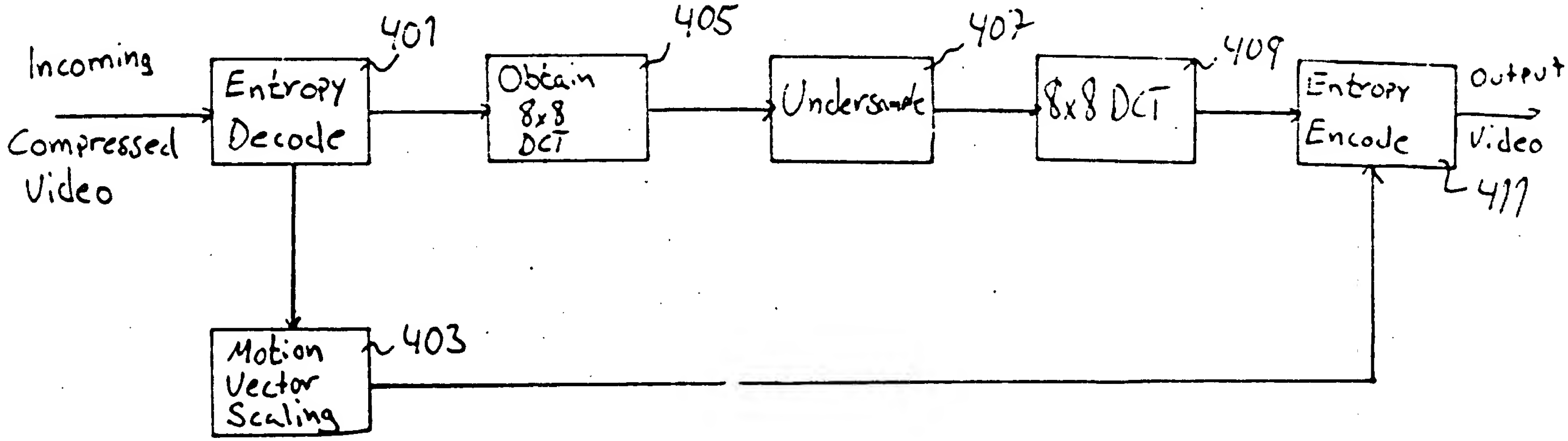

Fig. 4

0-9700026